

# MDS-Based Multiresolution Nonlinear Dimensionality Reduction Model for Color Image Segmentation

Max Mignotte

**Abstract**—In this paper, we present an efficient coarse-to-fine multiresolution framework for multidimensional scaling and demonstrate its performance on a large-scale nonlinear dimensionality reduction and embedding problem in a texture feature extraction step for the unsupervised image segmentation problem. We demonstrate both the efficiency of our multiresolution algorithm and its real interest to learn a nonlinear low-dimensional representation of the texture feature set of an image which can then subsequently be exploited in a simple clustering-based segmentation algorithm. The resulting segmentation procedure has been successfully applied on the Berkeley image database, demonstrating its efficiency compared to the best existing state-of-the-art segmentation methods recently proposed in the literature.

**Index Terms**—Berkeley image database, color textured image, multidimensional scaling, multiresolution optimization, nonlinear dimensionality reduction, probability rand index, unsupervised image segmentation.

## I. INTRODUCTION

**I**MAGE segmentation is an important preprocessing step which consists of dividing the image scene into spatially coherent regions sharing similar attributes. This low-level vision task, which changes the representation of an image into something that reduces the complexity of its content, is often the preliminary and also crucial step for higher level applications often designed to emulate functionalities of the human visual system (such as object localization or recognition, tracking, image retrieval, or understanding). Presently, the problem of finding a fast, simple, and automatic method that is able to efficiently segment a color textured image remains unresolved.

A number of methods have been proposed and studied in the last decades to solve the difficult problem of textured image segmentation. Among them, we can cite segmentation methods exploiting the connectivity information between neighboring pixels such as Markov random field (MRF)-based statistical methods [1]–[5] (or more generally Bayesian framework [6]), graph-based models [7]–[9], neural networks

[10], [11], variational methods [12], deformable surfaces [13], mean-shift-based techniques [14]–[16], clustering schemes [17]–[20] (with fuzzy sets [21] or Gaussian mixture models [22], [23]), color histograms [24], watershed techniques [25], region growing strategies [26], region-based split and merge procedures (sometimes directly expressed by a global energy function to be optimized [27]), and finally some fusion methods [17], [28]–[30].

Most of these above-mentioned methods preliminarily use (sometimes conjointly to the segmentation step) a texture feature extraction step whose goal is to characterize each meaningful textured region (to be segmented) with statistical (or geometrical, morphological, fractal, etc..) image features which are then either characterized by their distribution or simply gathered in a  $D$ -dimensional (feature) vector. When the structure of these features is simple enough and the dimensionality of the feature space is not too high (to be tractable), a segmentation model exploiting a feature likelihood using a mixture of Gaussians or other statistical distributions can be efficiently used. However, when the dimensionality of the classification problem is too high, a dimensionality reduction technique has to be preliminarily applied to get a manageable number of dimensions (avoiding the so-called “curse of dimensionality” problem). This dimensionality reduction step allows us to simplify (and to decrease the amount of time and memory required to perform) the subsequently used classification procedure (i.e., the clustering scheme or the final optimization problem) of the segmentation method [31]. In addition to decreasing the complexity of the clustering scheme or the final optimization problem, this dimensionality reduction step also allows us to remove redundancy and erroneous information from the data and sometimes to prevent the classification model from overfitting in the training phase. In this attempt, recent research in dimensionality reduction and segmentation of color images has focused on linear projection methods such as principal component analysis (PCA) [31] or independent component analysis (ICA) [32]. However, ICA and PCA are linear projection methods, and both assume that the underlying data manifold is linear, which is not necessarily true in the case of texture feature data. If the feature vectors are distributed along a nonlinear manifold embedded in a high-dimensional space, this dimension reduction step might lead to several classification errors.

Although the goals of segmentation/classification and dimensionality reduction are closely related, relatively little research has been conducted on the use of nonlinear

Manuscript received May 5, 2010; revised December 11, 2010; accepted December 15, 2010. Date of publication January 20, 2011; date of current version March 2, 2011. This work was supported in part by the Natural Sciences and Engineering Research Council under an individual research grant.

The author is with the Département d'Informatique et de Recherche Opérationnelle, Faculté des Arts et des Sciences, Université de Montréal, Montréal H3C 3J7, QC, Canada (e-mail: mignotte@iro.umontreal.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2010.2101614

dimensionality reduction and its benefit in segmentation of natural color images. However, in this attempt, a nonlinear dimensionality reduction was recently proposed in motion segmentation to deal with motion of different types [33] and in hyperspectral imagery [34] in order both to reduce the huge amount of high-dimensional spectral data information (related to each hyperspectral image) and to take into account its nonlinear characteristics (whose multiple sources of nonlinearity have been pointed out in [35]). To this end, Mohan *et al.* in hyperspectral imagery and Goh *et al.* [33] in motion segmentation have investigated the use of the locally linear embedding (LLE) algorithm [36] and its advantages in image segmentation. LLE attempts to represent the data manifold locally by reconstructing each feature vector as a weighted combination of its neighbors. Based on these affine (*local linear*) approximations, LLE finds a low-dimensional embedding of the data. In the same family, Laplacian eigenmaps (LEs) [37] and Hessian LLE (HLL) [38] have also been proposed as LLE techniques. Contrary to LLE, LE and HLL are based on computing the low-dimensional representation that best preserves *locality* instead of *local linearity* in LLE. A good review of these three algorithms is given in [33]. In a somewhat similar manner, it is also worth mentioning the curvilinear distance analysis (CDA) [39] and the isometric mapping (ISOMAP) [40]. Both begin by finding a set of the nearest neighbors of each point and then seek to preserve the geodesic distances between feature points while projecting the data into fewer dimensions. Geodesic distance is defined as the length of the shortest path between two points that stays on the manifold surface and can be computed using the Dijkstra's algorithm [41].

Another interesting nonlinear dimensionality reduction is based on multidimensional scaling (MDS) [42], which attempts to preserve pairwise distances between data points. More precisely, MDS attempts to find an embedding from the initial feature vectors in the high-dimensional space such that distances (usually the Euclidean distance but more generally, it may be a metric or arbitrary distance function) are preserved in a low-dimensional space. The foundational ideas behind MDS were first proposed by Young and Householder [43] and then further developed by Torgerson [44]. The MDS algorithm works by minimizing an objective function (called the *strain* or *stress* function) based on the discrepancy of these distances. The original algorithm, called classic MDS [44], exploits a spectral method which consists of finding embedding coordinates by computing the top eigenvectors of a "double-centered" transformation of the distance matrix sorted by decreasing eigenvalue. When a singular value decomposition is used for this purpose, the computational cost of classic MDS is of order  $O(N^3)$  (where  $N$  is the number of feature vectors), and when a more sophisticated method based on the power method is used to estimate the different eigenvectors, it is of order  $O(N^2)$ . It is also worth mentioning the majorization algorithm (called SMACOF) proposed by De Leeuw [45] which monotonically converges to a local minimum by minimizing a quadratic approximation at each iteration but also at a large computational cost of  $O(N^2d)$  ( $d$  being the dimensionality of the low-dimensional target space). In any

event, this nonlinear dimension reduction approach based on the original MDS algorithm requires too much computing when applied to all pixels of the image (especially when a high-dimensional vector is related to each pixel) and this certainly explains why such nonlinear reduction methods have not been further tested in image segmentation.

However, over the past 15 years, some strategies have been developed to allow the classical MDS algorithm to be applied for large-scale applications (i.e., for large sample size). To this end, three algorithms, i.e., FastMap [46], MetricMap [47], and Landmark MDS [48] (and its variant [49], [50]), have been proposed. The baseline strategy for these three algorithms is based on the Nyström approximation [51] of the eigenvectors of a large matrix (in this case, a Gramian matrix derived from the previously mentioned input distance matrix which costs  $O(N^3)$  complexity in time). The basic idea of the Nyström method can be deemed as choosing a subset of samples (approximating the eigensystem of this large distance matrix) to find a first nonlinear embedding and then extending the obtained solution to the complete set of remaining samples (using a mapping or a triangulation technique relative to the original data). Another approximation algorithm for MDS for use with large datasets is the FastMDS algorithm [52]. FastMDS exploits a divide-and-conquer approach by first dividing the large source matrix into submatrices of reasonable size to perform the classical MDS algorithm (on each of them) and then by stitching the subproblem solutions back together for a complete solution (with an affine mapping in a least-squares sense). All these above-mentioned methods chose the strategy of transforming the initial nonlinear optimization problem, i.e., the minimization of the so-called stress function (proportional to the distortion of all pairwise distances) into a linear algebra problem of eigendecomposition along with a Nyström sampling or a divide-and-conquer scheme that facilitates the eigendecomposition. To our knowledge, none of them tackles this scalable MDS task directly as an optimization problem and uses a (multiresolution) subsampling and interpolation schemes which exploit the important features of a (textured) natural image, i.e., the inherent spatial dependencies between spatial neighboring samples (and modeling the fact that for a sample or a texture feature vector belonging to a particular region, its spatial neighboring samples likely belong to the same region). In our application, these inherent spatial dependencies between (spatial) neighboring samples allow us to define a multiresolution representation of the data in order to construct approximate coarser versions of the original (nonconvex and difficult) optimization problem and to use the solution of the coarser (and hence computationally simpler) version of this optimization to obtain a good initial guess that guides and accelerates the solution of finer versions.

Scalable MDS (or projection) methods are also useful techniques for (text) data mining. In particular, these dimensionality reduction methods can create a visual and logical representation of a large data document set (based on the word frequency) in a low-dimensional space that is most expressive and thus leads to an easier interpretation of a large text corpus. In this field of research, Chen *et al.* [53] have proposed a scalable dimensionality reduction technique which

exploits both a Nyström approximation-based method and, in a somewhat similar way to our strategy, relevant information about the latent structure in the given data. More precisely, they first select a subset of representative exemplars (summarizing the entire dataset) in order to generate a first nonlinear embedding (in the low-rank space) on which they obtain a “soft” classification, representing the class or the topic of each document in the document collection. The following step of their method is to use this soft clustering and a parametric embedding to extend to the remaining documents. In this strategy, the “soft” classification, which concretely represents the empirical probability of each document to the different topics, turns out to be more efficient than the class of hierarchical-clustering-based dimensionality reduction techniques [53] usually proposed as large-scale document visualization methods and which use a first “hard” classification (and for which an incorrect assignment made during the classification step can lead to error in the final text mining results).

In this paper, we present an efficient coarse-to-fine multiresolution framework for the MDS technique especially suited for large-scale dimensionality reduction problems such as those occurring during the texture feature extraction step in (texture) image segmentation or (more generally) occurring in large-scale high-dimensional clustering applications (e.g., bioinformatics). In this context, we demonstrate the efficiency of such a technique and its real interest to learn a nonlinear low-dimensional representation of the texture feature data of a natural image, which will be then efficiently exploited in a segmentation procedure. As is usually the case in such a coarse-to-fine strategy, our multiresolution MDS technique estimates approximate coarser versions of the original optimization MDS problem (i.e., the minimization of the so-called strain objective function) and then uses the solution of the coarser (and hence computationally simpler) estimate of this optimization to obtain (via an interpolation scheme) a good initial guess that guides and accelerates the optimization of finer versions. In our application, in order to obtain an accurate solution at each scale, we have designed a hybrid (but simple) relaxation scheme combining a global and deterministic gradient-based procedure with a local stochastic search, which ultimately allows us to refine the estimation previously given by the deterministic relaxation technique.

The remainder of this paper is organized as follows. Section II describes the texture feature extraction step. Sections III and IV present respectively the MDS-based reduction model and our multiresolution optimization framework, finally, Section V presents a set of experimental results and comparisons with existing segmentation techniques.

## II. TEXTURE FEATURE EXTRACTION STEP

In order to validate our segmentation model, we use the following as texture features (to characterize each textured region) and to be represented in a lower dimensional space.

- 1) First, the set of values of the local color non-normalized histogram and estimated around the pixel to be classified for an input image expressed in the LAB color space. In our application, this local histogram is equally re-quantized with  $q_c$  equidistant binnings for each of the

A and B color channels, and  $(q_c - 1)$  equidistant binnings for the L channel (in order to be somewhat invariant to shadows effects) in a final  $N_b = (q_c - 1) \times q_c^2$  bin descriptor, computed on an overlapping squared fixed-size ( $N_w$ ) neighborhood centered around the pixel to be classified.

- 2) Second, the set of respectively vertical, horizontal, right diagonal, and left diagonal different equidistant values of the non-normalized local histogram of the gradient magnitude (i.e., the absolute value of the first-order difference) computed on the luminance component of each pixel contained in an overlapping squared fixed-size ( $N_w$ ) neighborhood centered around the pixel to be classified. In our application, we use the four sets of  $N_g$  bin values in the interval  $[0 : G_{\max}]$  of this four (i.e., vertical, horizontal, right diagonal, and left diagonal) gradient magnitude histogram.

This simple texture feature extraction step thus yields a  $[N_b + 4 N_g] = [(q_c - 1)q_c^2 + 4 N_g]$ -dimensional feature vector (for each pixel) whose dimension will be nonlinearly reduced by our multiresolution MDS model (in our application  $q_c = 4$ ,  $N_g = 10$ , and  $N_w = 7$ ). In this simpler model, a *texton* (i.e., the repetitive character or element of a textured image, also called a texture primitive) is thus characterized here by the values of two types of local histograms: first, the color histogram [29] and, second, the gradient magnitude histogram in the four directions (see Algorithm 1 for implementation details).

## III. PROPOSED REDUCTION MODEL

Since we have two different types of texture features (i.e., color and gradient) to characterize a textured region, and since there is no reason why these two texture clues are interrelated, we separately reduce the dimensionality of the color and the gradient magnitude feature vectors. This strategy allows the MDS algorithm to more easily find a nonlinear manifold (or equivalently, to considerably simplify the optimization procedure). Moreover, since the color features seem more important than the gradient magnitude clues [17] to characterize a texture region, we also use twice as much weight (compared to the gradient feature vectors) by searching for them in a nonlinear manifold with two times more dimensions. Finally, we construct a low-dimensional representation with three dimensions; this allows us to visualize this low-dimensional representation as a three-channel (e.g., RGB) image. In this context, we have therefore a space with a dimensionality of 2 available for the color vector and a space with a dimensionality of 1 for the gradient magnitude vector.

Let us define the notation that will be used throughout this paper. We consider the set of texture features of an input color image (with  $N$  pixels) as a cube  $\mathcal{I}(\mathbf{s}, k)$  or a 3-D array, where  $\mathbf{s}$  indicates the spatial location  $\mathbf{s} = (\text{row}, \text{column}) = (i, j)$  and  $f_{\mathbf{s}}(k) = \mathcal{I}(\mathbf{s}, k) \in \mathbb{R}^D$  is the texture feature vector, indexed by  $k$ , at location  $\mathbf{s}$ .  $D$  is the dimensionality of the original texture feature vectors (i.e., the high-dimensional space) and  $d = 3$ , the dimensionality of the target low-dimensional representation. In this context, the MDS approach, which refers to the optimization of these  $N$  texture feature vectors

**Algorithm 1** Estimation of the Bin Descriptor for Each Pixel  $s$ 

$N_w$	Size of the local window
$q_c$	Bin number of the color histogram (A&B channels)
$q_{c-1}$	Bin number of the color histogram (Lchan.)
$G_{max}$	Maximal value of the gradient histogram
$N_g$	Number of bin values for each of the four (hor., vert., right and left diag.) gradient histograms
$s$	Pixel spatial location $s = (\text{row}, \text{col}) = (i, j)$
$\mathcal{N}_s$	Set of pixel locations $s$ within the $N_w \times N_w$ neighborhood region centered at $s$
$h[]$	Bin descriptor : array of $D = ((q_c - 1) \cdot q_c^2 + 4 N_g)$ integers ( $h[0], h[1], \dots, h[D-1]$ ) initialized to 0
$\lfloor \cdot \rfloor$	Integer part of.
$q_g$	Bin number of the gradient histogram

**Initialization**

$$q_g \leftarrow G_{max} / N_g$$

**Color histogram**

for each pixel  $s \in \mathcal{N}_s$  with color value  $L_s, A_s, B_s$  do

$$\begin{aligned} & \circ k \leftarrow q_c^2 \cdot \lfloor \frac{(q_c - 1) \cdot L_s}{256} \rfloor + q_c \cdot \lfloor \frac{q_c \cdot A_s}{256} \rfloor + \lfloor \frac{q_c \cdot B_s}{256} \rfloor \\ & \bullet h_s[k] \leftarrow h_s[k] + 1 \end{aligned}$$

**Edge histogram**

for each pixel  $s \in \mathcal{N}_s$  with Luminance value  $L_s$  do

$$\begin{aligned} & \circ k \leftarrow \lfloor \frac{|L_{(i-1,j)} - L_{(i+1,j)}|}{q_g} \rfloor + (q_c - 1) \cdot q_c^2 \\ & \bullet h_s[k] \leftarrow h_s[k] + 1 \\ & \circ k \leftarrow \lfloor \frac{|L_{(i,j-1)} - L_{(i,j+1)}|}{q_g} \rfloor + (q_c - 1) \cdot q_c^2 + N_g \\ & \bullet h_s[k] \leftarrow h_s[k] + 1 \\ & \circ k \leftarrow \lfloor \frac{|L_{(i-1,j-1)} - L_{(i+1,j+1)}|}{q_g} \rfloor + (q_c - 1) \cdot q_c^2 + 2N_g \\ & \bullet h_s[k] \leftarrow h_s[k] + 1 \\ & \circ k \leftarrow \lfloor \frac{|L_{(i-1,j+1)} - L_{(i+1,j-1)}|}{q_g} \rfloor + (q_c - 1) \cdot q_c^2 + 3N_g \\ & \bullet h_s[k] \leftarrow h_s[k] + 1 \end{aligned}$$

$\in R^D$  in our target space whose dimensionality is  $d = 3$ , using a separate optimization between the color (2-D) and gradient features (1-D) in such a way that their distance relationships in the target space faithfully reflect those of the original texture feature vectors, consists of finding  $\hat{\mathbf{u}} = (\hat{R} \hat{G})^t$  a 2-D vector of mappings and  $\hat{\mathbf{v}} = (\hat{B})^t$  a 1-D vector solution of the two (independent and) objective *stress* functions to be minimized

$$\begin{cases} \hat{\mathbf{u}} = \arg \min_{\mathbf{u}} \sum_{s, t, s \neq t} \left( w_{s,t} \beta_{s,t}^{[0:N_b]} - (\mathbf{u}_s - \mathbf{u}_t)^2 \right)^2 \\ \hat{\mathbf{v}} = \arg \min_{\mathbf{v}} \sum_{s, t, s \neq t} \left( w_{s,t} \beta_{s,t}^{[N_b:D]} - (\mathbf{v}_s - \mathbf{v}_t)^2 \right)^2 \end{cases} \quad (1)$$

where  $\beta_{s,t}^{[k_0-k_1]}$  denotes the squared distance between the pair of feature vectors  $f_s(k)$  and  $f_t(k)$  at pixels locations  $s$  and  $t$  and  $k \in [k_0 - k_1]$ . The summation  $\sum_{s, t, s \neq t}$  is over all pairs of sites (i.e., for all sites  $s$  and for all pairs of sites including  $s$ )

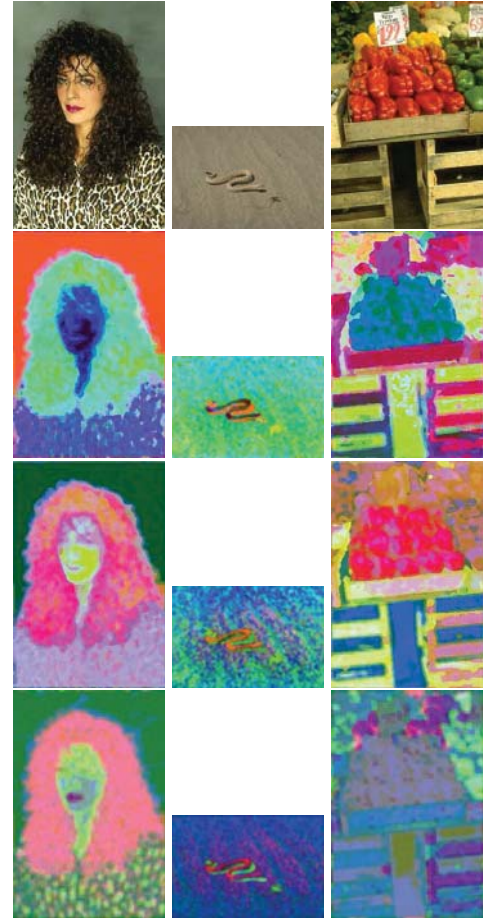


Fig. 1. Top row: Original Berkeley images ( $n^0$  198054 and  $n^0$  196073 and  $n^0$  25098) and detextured related images i.e., low 3-D representation (as a color image) of the local color value distribution of the texture regions with. Second row: our multiresolution MDS algorithm (with respectively, from left to right,  $\text{RMSRDE} = 0.901, 0.919, \text{ and } 0.916$ ). Third row: PCA technique. Fourth row: FastMap algorithm (with respectively, from left to right,  $\text{RMSRDE} = 0.957, 0.870, \text{ and } 0.951$ ).

existing in the image and we recall that  $N_b = (q_c - 1)q_c^2$ .  $w_{s,t}$  is a factor equal to 0.5 for sites  $t$  belonging to the first neighborhood of  $s$ . This allows us to easily include in this dimension reduction step a prior favoring homogeneity between neighboring sites. In our application, the three mapping results ( $\hat{R} \hat{G} \hat{B}$ ) can be seen as a (noisy) detextured color image (see Fig. 1).

An important ingredient of the MDS algorithm is the choice of the distance measure which will be used in the above-stated optimization problem. In our context, one benefit of the MDS method over the other nonlinear dimensionality reduction models is its ability to be very flexible in the choice of this distance measure. By comparison, LLE and its variants (LE, HLLE) involve writing each data point as a linear combination of its neighbors and, in the Euclidean case, this step is simply a least-squares problem, but a computationally costly problem for other distance measures. Moreover, there exists nonlinear dimension reduction strategies in which this distance measure is fixed once and for all because it is intrinsically part of the nonlinear reduction model (e.g., CDA and ISOMAP). Our application (a better choice than the Euclidean distance, since we use histogram values as texture features)

includes a histogram-based similarity measure derived from the Bhattacharyya similarity coefficient. Given one normalized histogram  $h_s^*(k)$  at pixel location  $\mathbf{s}$  and another  $h_t^*(k)$  at location  $\mathbf{t}$ , the squared Bhattacharyya distance between these two histograms is defined as

$$\beta_{\mathbf{s},\mathbf{t}}^{\mathcal{I}} = D_B[h_s^*, h_t^*] = 1 - \sum_{k \in \mathcal{I}} \sqrt{h_s^*(k) h_t^*(k)} \quad (2)$$

in which the resulting squared distance ranges from 0 to 1. It is also worth mentioning that each one of the three objective functions to be minimized [see (1)] involves nonlinear non-convex objective functions, thus requiring high computations. Moreover, it is also worth noting that each function to be optimized can be viewed as a Gibbs energy field related to a non-stationary (and nonlocal) MRF model defined on a *complete* graph with long-range pairwise interactions, i.e., binary cliques  $\langle \mathbf{s}, \mathbf{t} \rangle$  (or pairwise of pixels). In this context, a stochastic optimization procedure such as a simulated annealing (SA) algorithm [1] has the capability of avoiding local *minima* but, on the other hand, is computationally too expensive. Another approach is to use a classical gradient descent algorithm or a deterministic variant of the SA procedure, namely the iterative conditional modes introduced by Besag [54]. This method, which is simply a gradient descent alternating the directions, i.e., that selects a variable while keeping all other variables fixed, is deterministic and simple. Nevertheless, it requires a good initialization (close to the optimal solution). Otherwise, it will converge toward a (very) bad local minimum. Keep in mind that each of the objective functions of our dimensionality reduction model is of the form  $\sum_{\mathbf{s}, \mathbf{t} \neq \mathbf{t}} V_{\mathbf{s},\mathbf{t}}(\cdot)$  with nonconvex and nonmetric or semimetric potential functions  $V_{\mathbf{s},\mathbf{t}}(\cdot)$  (since in our case  $\mathbf{u}_{\mathbf{s}} = \mathbf{u}_{\mathbf{t}} \not\Rightarrow V_{\mathbf{s},\mathbf{t}}(\mathbf{u}_{\mathbf{s}}, \mathbf{u}_{\mathbf{t}}) = 0$ ). Consequently, (1) does not belong to the class of energy functions that can be minimized via graph cut techniques such as the expansion-move and the swap-move algorithms. This is especially true because  $V_{\mathbf{s},\mathbf{t}}$  clearly does not satisfy the (necessary) condition of regularity [55, Sec. 4], which has more relaxed applicability conditions than required to get a graph representability for the energy minimization by graph cuts. Optimization algorithms such as loopy belief propagation (LBP) [56] or its variant, the so-called tree-reweighted message passing algorithm [57], are more general and can be applied to any type of potential function. However, in our case these optimization techniques would be very slow and expensive in terms of memory requirement. In these optimization strategies, each node sends a (different) message (in fact an integer value) to each of its neighbors and receives a message from each neighbor (at each iteration). For our MRF model defined on a complete graph, there would be  $O(N^2)$  messages (to compute and store) per iteration (with  $N$ , the number of pixels of an image). Moreover, as standard way of computing the messages is to explicitly minimize over  $\mathbf{u}_{\mathbf{s}}$  for each choice of  $\mathbf{u}_{\mathbf{t}}$  [56], it would take  $O(k^2)$  time to compute each message with  $k = 256$ , i.e., the number of existing discrete color channel values. The standard implementation of these message-passing algorithms, on our (complete) graph, would thus require a prohibitive computational complexity of order  $O(N^2 k^2 T)$  with  $T \approx N^{1/2}$  (the number of iterations of the LBP needs to grow like  $N^{1/2}$

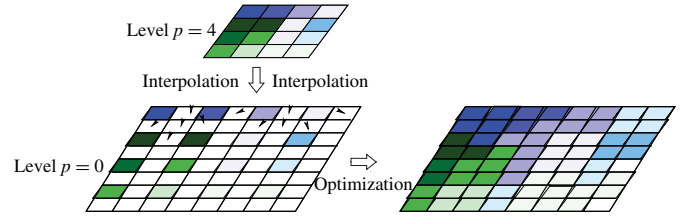


Fig. 2. Interpolation and “coarse-to-fine” optimization strategy.

[56] to allow for information from one part of the image to propagate everywhere else). In addition to this, LBP is not guaranteed to find a global minimum, but only a strong local minimum [58]. Moreover, LBP is also not guaranteed to converge since it may go into an infinite loop switching between two labelings [58]. Note also that we cannot consider a dynamic programming approach [59] since this optimization method is restricted essentially to energy functions in 1-D settings.

An alternative approach consists of a multiresolution optimization strategy. In this coarse-to-fine strategy, rather than considering the minimization problem on the full and original configuration space, the original optimization problem is decomposed into a sequence of approximated optimization problems of reduced complexity. This drastically reduces computational effort and provides an accelerated convergence toward an improved estimate (experimentally, estimation results are nearly comparable to those obtained by stochastic optimization procedures as noted, for example, in [60] and [61] and in the experimental results of this paper below).

#### IV. MULTIREOLUTION OPTIMIZATION STRATEGY

##### A. Principle

In order to decrease the computational load of our coarse-to-fine optimization procedure, we only use two levels of resolution in our multiresolution optimization strategy: namely, the full resolution  $p = 0$  and a coarser (image) solution at resolution level  $p = 4$ , i.e., resulting from the downsampling of the input image and the (low-dimensional embedding image) solution  $\hat{\mathbf{u}}^{\downarrow 0} \equiv \hat{\mathbf{u}}$  by  $2^{(p=4)} = 16$  in each direction (if we consider the first mapping  $\hat{\mathbf{u}}$  to be estimated). At these two levels of resolution, we also consider that each image is assumed to be toroidal, i.e., periodically repeated. At the lower resolution level, we consider that each node (or pixel) is connected with all other pixels located within a square neighborhood window of fixed size  $N_s = 15$  pixels centered around the pixel. This allows us to consider nearly a complete graph for a Berkeley image of size  $320 \times 240$  such as an image belonging to the Berkeley database (once downsampled by a factor  $2^4$ ). At the finest resolution level, we consider the same square neighborhood window. After convergence our optimization scheme at coarser level, the result obtained is interpolated and then used as initialization for the full resolution level (see Fig. 2).

##### B. Interpolation

For the interpolation at the full resolution level, we have used an interpolation method which efficiently takes into



**Algorithm 2** Local Exploration with Metropolis

---

$E$	Energy function to be minimized
$T_l$	Temperature at Iteration step $l$
$a$	Cooling schedule parameter
$r$	Radius of exploration, real $\epsilon \in ]0, 1]$
$T_0$	Initial Temperature
$T_f$	Final Temperature
$L_{\max}^s$	Maximal number of iterations

1. **Initialization**  $a \leftarrow \left( \frac{T_f}{T_0} \right)^{\frac{1}{L_{\max}^s}}$
2. **Local Exploration**  
**while**  $l < L_{\max}^s$  **do**
  - for each pixel with value  $x_s$  at site  $s$  do**
    - Compute  $\Delta \text{Energy} = E(y_s) - E(x_s)$  with  $y_s \in [x_s - 255r : x_s + 255r]$  and  $y_s$  (pixel value)  $\in [0 : 255]$
    - **If** ( $\Delta \text{Energy} < 0$ ) Replace  $x_s$  by  $y_s$
    - **Else** Replace  $x_s$  by  $y_s$  with probability  $\triangleright \exp\left(-\frac{\Delta \text{Energy}}{T_l}\right)$
3.  $l \leftarrow l + 1$  and  $T_l \leftarrow T_0 a^l$

---

account three characteristics:

- 1) the non-reduced (complete) data for each pixel  $s$ , i.e.,  $\mathcal{I}(s, \cdot)$ ;
- 2) the inherent spatial dependencies between neighboring texture feature vectors (i.e., modeling the fact that for a given texture feature vector belonging to a particular region or class, its surrounding pixels likely belong to the same class);
- 3) the weighted average formula of the nonlocal means filtering strategy used in [62] which is very robust to noise.

More precisely, in our duplication method, each value of a mapping to be interpolated, at site  $s$  and at full resolution level  $p = 0$  of  $\hat{\mathbf{u}}$ , is computed as a weighted average of all the values in the neighborhood of  $s$  at level resolution  $p = 4$  by the following average formula:

$$\hat{\mathbf{u}}_s^{\downarrow 0} = \sum_{\mathbf{t} \in \mathcal{N}_s} w(\mathbf{s}, \mathbf{t}) \hat{\mathbf{u}}_s^{\downarrow 4} \quad (3)$$

where  $\mathcal{N}_s$  designates a square fixed-size ( $N_l$ ) neighborhood centered around the pixel  $s$ . In this interpolation formula, the weights  $\{w(\mathbf{s}, \mathbf{t})\}_{\mathbf{t}}$  depend on the similarity (according to the Euclidean distance) between the texture feature vector  $\mathcal{I}(s, \cdot)$  and  $\mathcal{I}(t, \cdot)$

$$w(\mathbf{s}, \mathbf{t}) = \frac{1}{Z(\mathbf{s})} \exp \left\{ -\frac{\|\mathcal{I}(\mathbf{s}, \cdot) - \mathcal{I}(\mathbf{t}, \cdot)\|_2^2}{h} \right\} \quad (4)$$

where  $Z(\mathbf{s})$  is the normalizing constant ensuring  $\sum_{\mathbf{t}} w(\mathbf{s}, \mathbf{t}) = 1$ . The parameter  $h$  acts as a degree of filtering, it controls the decay of the weights as a function of the Euclidean distance. In our application, we use  $h = 0.1$  and  $N_l = 5$ .

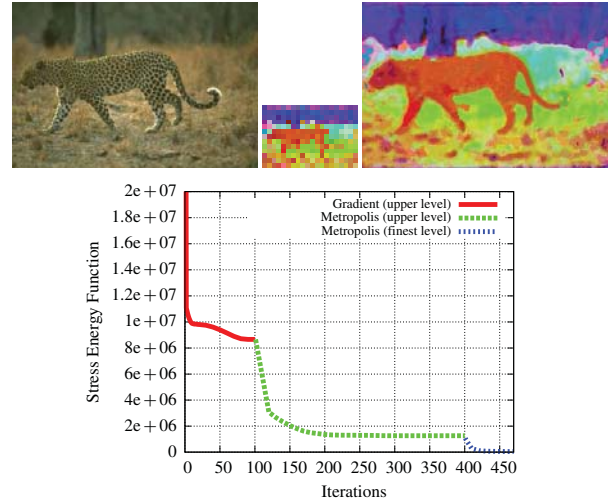


Fig. 3. Multiresolution optimization strategy on a natural image from the Berkeley database ( $n^0$  134052). From left to right, the (three-band) color image at coarser resolution level (initialization at random), and nonlinear dimensionality reduction result obtained after convergence of the conjugate gradient [represented as a (three-band) color image], duplication and result of our hybrid algorithm at the finest level of the pyramid, i.e., at full resolution. Bottom: evolution of the stress energy function for the two steps of the multiresolution hybrid optimization scheme (for the color features).

### C. Optimization at Coarser and Finest Resolution Levels

- 1) Since an analytical expression of the derivative of the objective function to be optimized can be easily found, we first use a conjugate gradient descent procedure with derivative

$$\nabla E(\mathbf{u}) = 4 \sum_{\mathbf{s}, \mathbf{t} \neq \mathbf{t}} [\mathbf{u}_s - \mathbf{u}_t] \left( \rho_{s, \mathbf{t}}^{[k_0 - k_1]} - [\mathbf{u}_s - \mathbf{u}_t]^2 \right). \quad (5)$$

For the search of the optimal mapping at the coarser level, our optimization procedure is initialized at random. For the gradient descent, the step size is fixed to  $\gamma$  and adaptively decreased by a factor of 2 if the energy to be minimized increases between two iterations. We stop the optimization procedure if a fixed number of iterations is reached.

- 2) In order to refine the estimation given by the above-mentioned deterministic optimization method at coarser level or to refine the estimation at full resolution (after the interpolation [see Section IV-B]), we use the previous optimization result as the initialization of a stochastic local search. To this end, we use a local exploration around the current solution using the Metropolis criteria [63] and a low radius of exploration (see Algorithm 2).

Details of our multiresolution optimization strategy with two levels of resolution is given in Algorithm 3.

## V. EXPERIMENTAL RESULTS

### A. Image Segmentation

The above-presented nonlinear dimension reduction step can be viewed as a detexturing approach which converts the original texture image into a noisy image with regions with uniform colors in a low-dimensional space. This detexturing approach simplifies greatly our segmentation problem

**Algorithm 3** Multiresolution Minimization

$I(s, k)$  Input texture image  
 $I^{\downarrow p}$   $I(s, k)$  downsampled by a factor of  $2^p$   
 $\{\beta_{s,t}\}^{\downarrow p}$  Set of distances between  $I(s, \cdot)$  and  $I(t, \cdot)$  obtained from  $I^{\downarrow p}$   
 $\mathbf{u}^{\downarrow p}$  mapping (R or G or B) to be estimated at resolution level  $p$  ( $\mathbf{u}^{\downarrow 0} \equiv \mathbf{u}$ )  
 $[b_1 - b_2]$  Interval of the texture feature index  
 $p = 4$  Upper level of the pyramidal structure

**1. Initialization Step**

- $I^{\downarrow p} \leftarrow$  Downsampled  $I(s, k)$  by a  $2^p$  factor
- $\{\beta_{s,t}\}^{\downarrow p} \leftarrow$  Compute the set of distances from  $I^{\downarrow p}$
- $\{\beta_{s,t}\}^{\downarrow 0} \leftarrow$  Compute  $\{\beta_{s,t}\}$  from  $I^{\downarrow 0} = I$

**2. Upper level**

- $\mathbf{u}^{\downarrow p} \leftarrow$  Random
- $\mathbf{u}^{\downarrow p} \leftarrow$  HYBRID OPTIMIZATION ( $\mathbf{u}^{\downarrow p}$ ,  $\{\beta_{s,t}^{[b_1-b_2]}\}^{\downarrow p}$ )
- $\hat{\mathbf{u}} \leftarrow$  Interpolation ( $\hat{\mathbf{u}}^{\downarrow p}$ )

**3. Finer level**

- $\hat{\mathbf{u}} \leftarrow$  METROPOLIS SEARCH ( $\hat{\mathbf{u}}$ ,  $\{\beta_{s,t}^{[b_1-b_2]}\}$ )

(see Figs. 1 and 3). In order to demonstrate this, we can first compare our detexturing approach with the PCA [64] method,<sup>1</sup> which is also the most classical linear dimensionality reduction techniques also known as Karhunen–Loeve transformation.

As in our model, we have also used a 2-D mapping for the color-based features and a 1-D mapping for the gradient based features. Fig. 1 shows the low 3-D representation (as a color image) of the local texture regions with the MDS and the PCA techniques (the FastMap-based results will be discussed in Section V-E). We can easily see that the PCA-based detexturing approach converts the different textural regions of the original image with: 1) less uniform (or more noisy) colors (e.g., the snake image or the girl's face); 2) with less dissimilar colors (e.g., the hair and the sweater of the girl); and/or 3) uses sometimes wrongly similar colors for two different textured regions (e.g., the green peppers and the background region composed of white cabbages and salads). In our opinion, our MDS-based detexturing method gives better detextured images compared to the PCA technique (and thus more effective detextured images for the subsequent use of a clustering algorithm (see Section V-C and Table I) for two main reasons. The first one is, of course, due to the nonlinearity of our dimensionality reduction technique. The second one is also certainly due to the nonlinear Bhattacharyya distance which is inherent to our dimensionality reduction model, whereas the PCA technique is inherently related to the linear Euclidean distance (since PCA yields linear transformation with minimum mean square error (or variance) which is defined in terms of second powers just like the Euclidean distance metric [65]).

<sup>1</sup>Due to the size of the covariance matrix, we use the same multiresolution technique, i.e., the estimation of the covariance matrix on the 16 times smaller image (corresponding to the fourth upper level of a multiresolution pyramidal structure) and our interpolation scheme described in Section IV-B.

TABLE I  
AVERAGE PERFORMANCE, IN TERM OF PROBABILITY RAND INDEX (PRI)  
MEASURE, OF OUR ALGORITHM FOR SEVERAL VALUES OF TTs  
INTERNAL PARAMETERS ON THE BERKELEY IMAGE DATABASE [66]

ALGORITHMS	PRI [67]
HUMANS (in [19])	0.875
<b>MD2S</b> <sup>K-means</sup> [K = 11   $\xi = 0.4$ ]	<b>0.784</b>
<b>MD2S</b> <sup>Mean-Shift</sup> [ $\sigma_s = 27$   $\sigma_r = 55$   $\xi = 0.4$ ]	<b>0.786</b>
<b>FASTMAP</b> <sup>K-means</sup> [K = 8   $\xi = 0.18$ ]	<b>0.755</b>
<b>PCA</b> <sup>K-means</sup> [K = 6   $\xi = 0.17$ ]	<b>0.730</b>
-2010- PRIF [29]	0.801
-2008- CTex [18]	0.800
-2009- MIS [13]	0.798
-2008- FCR [17]	0.788
-2004- FH [8] (in [19])	0.784
-2009- HMC [3]	0.783
-2009- Consensus [30]	0.781
-2009- A-IFS HRI [21]	0.771
-2001- JSEG [26] (in [18])	0.770
-2007- CTM [19]	0.762
-2008- St-SVGMM [22]	0.759
-2003- Mean-Shift [4] (in [19])	0.755
-2008- NTP [9]	0.752
-2010- iHMRF [4]	0.752
-2006- GBMS [16] (in [6])	0.734
-2000- NCuts [7] (in [19])	0.722
-2010- JND [24]	0.719
-2010- DCM [6]	0.708
-2009- [25]	0.703

In addition, we can now use this MDS-based detextured image as input image of a segmentation algorithm. In our experiments, we have tested the following.

- 1) A simple  $K$ -means clustering procedure. To this end, we have used as input multidimensional feature descriptor the set of values estimated on an overlapping squared fixed-size ( $N_{km} = 5$ ) neighborhood centered around each pixel (of our detextured image) to be classified and a number of classes ( $K = 11$ ) learned on a training image database (see Section V-C). A final merging step is added to each segmentation map, which simply consists of fusing each small region (i.e., regions whose size is below 300 pixels) with the region sharing its longest boundary.
- 2) A mean-shift [14] procedure whose internal parameters are learned with our grid-search-based learning method (see Section V-C).

In order to further help the  $K$ -means clustering process or the mean-shift procedure to succeed in finding an accurate partition, a simple hard constraint taking into account the edge of our detextured image and/or enforcing the spatial continuity of each (likely) region is imposed during the iterative  $K$ -means labeling process. To this end, we have to compute an edge gradient magnitude map from our detextured image. The most likely regions in this edge map (see Fig. 4) are easily estimated by identifying the sets of connected pixels

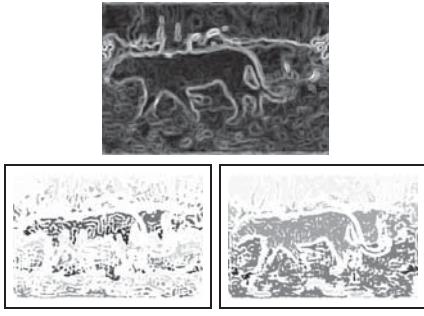


Fig. 4. From top to bottom and left to right: Edge gradient magnitude map of the detextured image presented in Fig. 3 and the sets of connected pixels (i.e., regions) whose edge potential is respectively below  $\xi = 0.3$  and  $\xi = 0.4$ . These regions are represented by colored regions. The white region corresponds to the sets of pixels whose edge potential is above the threshold (and thus corresponds to inhomogeneous regions).

whose edge potential is below a given (and low) threshold  $\xi$ , thus defining a map of likely homogeneous regions (in which only small gradient magnitudes have been detected). The hard constraint enforcing the spatial continuity of each of the  $K$ -means cluster is then simply performed by assigning the majority class label in each (pre-estimated) homogeneous textural region, for each iteration of the  $K$ -means algorithm (or at the end of the mean-shift procedure). This spatially constrained  $K$ -means is ensured after convergence of the classical  $K$ -means clustering. Implicitly, this procedure allows the consideration of nonspherical clusters in the  $K$ -means clustering scheme since the distribution of each textural feature is no longer spherical after the spatial constraint.

### B. Sensitivity to Internal Parameters

The two internal parameters of our segmentation algorithm are thus  $K$ , i.e., the number of classes of  $K$ -means, and our regularization parameter  $\xi$  (see the previous section). The other parameter belongs to the feature extraction step and the MDS-based dimensionality reduction technique. In this regard, concerning the MDS technique.

- 1) It is worth mentioning that our algorithm is relatively insensitive to high values of the step size  $\gamma$  because of our adaptive decreasing schedule which adaptively adjusts and reduces this value in the conjugate gradient procedure if this parameter is mistakenly set too high.
- 2)  $T_f$  is easily findable in our case, since a good final temperature for a Metropolis-like minimization procedure has to ensure that, at the end of the stochastic search, very few sites change their luminance values between two complete image sweeps. In our algorithm, this parameter has been easily found after a few trials.
- 3) Finally, two internal parameters are sensitive and crucial for our MDS algorithm: namely, the radius of exploration  $r$  and in a least measure, the starting temperature  $T_0$  of the local stochastic search. The first parameter was set in order to locally explore a solution whose luminance values are close to the solution given by the gradient minimization procedure (a value  $r = 0.04$  ensures that the final solution will exhibit output luminance values centered around the gradient estimation

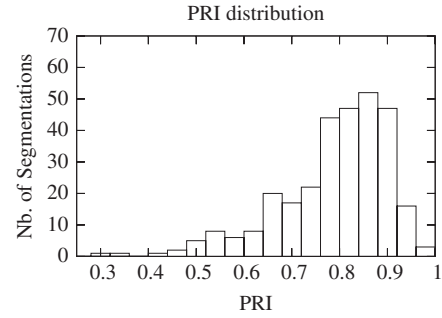


Fig. 5. Distribution of the PRI performance measure over the 300 images of the Berkeley database (for MD2S $^{K\text{-means}}_{[K=11|\xi=0.40]}$ ).

$\pm 0.04 * 255 = \pm 10$  luminance values for a final luminance image whose luminance values are comprised in  $[0 : 255]$ ).  $T_0$  is set in order to ensure that, at the beginning of the stochastic search, approximately 50% of sites change their luminance values between two complete image sweeps.

### C. Setup

In these experiments, we have tested our segmentation algorithms on the Berkeley segmentation database [66] consisting of 300 color images of size  $481 \times 321$  divided into a training set of 200 images and a test set of 100 images. For each color image, a set of benchmark segmentation results, provided by human observers (between 4 and 7), is available and used to quantify the reliability of the proposed segmentation algorithm. In order to ensure the integrity of the evaluation, the internal parameters of the MDS +  $K$ -means algorithm ( $K$  and  $\xi$ ) are tuned on the trained image set by doing a local discrete grid-search routine, with a fixed step size, on the parameter space and in the feasible ranges of parameter values (namely  $K \in [3 - 15]$  [step-size: 1],  $\xi \in 255 \times [0 - 1]$  [step-size: 0.05]). The internal parameters of the MDS + mean-shift (i.e., the two parameters controlling the resolution in the spatial and range domains and  $\xi$ ) and the PCA +  $K$ -means algorithms are learned by our grid-search routine in the same way. The algorithm is then benchmarked by using the optimal training parameters on the independent test set. The PRI [67] result is then given for the entire image database for comparison with the other segmenters (see Table I). In our algorithm, all color images are normalized to have the longest side equal to 320 pixels. The segmentation results are then supersampled in order to obtain segmentation images with the original resolution ( $481 \times 321$ ) before the estimation of the performance metric. We have compared our segmentation algorithms called MD2S (for MDS-based segmentation), namely, MD2S $^{K\text{-means}}_{[K=11|\xi=0.4]}$  and MD2S $^{\text{mean-shift}}_{[\sigma_s=27|\sigma_r=55|\xi=0.4]}$  against several unsupervised algorithms. For each of these algorithms, the internal parameters are set to optimal values and/or correspond to the internal values suggested by the authors. The comparison is based on the PRI<sup>2</sup> performance

<sup>2</sup>We have used the MATLAB code proposed by A. Y. Yang in order to estimate the PRI performance measure presented in the following section. This code is available online at [http://www.eecs.berkeley.edu/~yang/software/lossy\\_segmentation/](http://www.eecs.berkeley.edu/~yang/software/lossy_segmentation/).



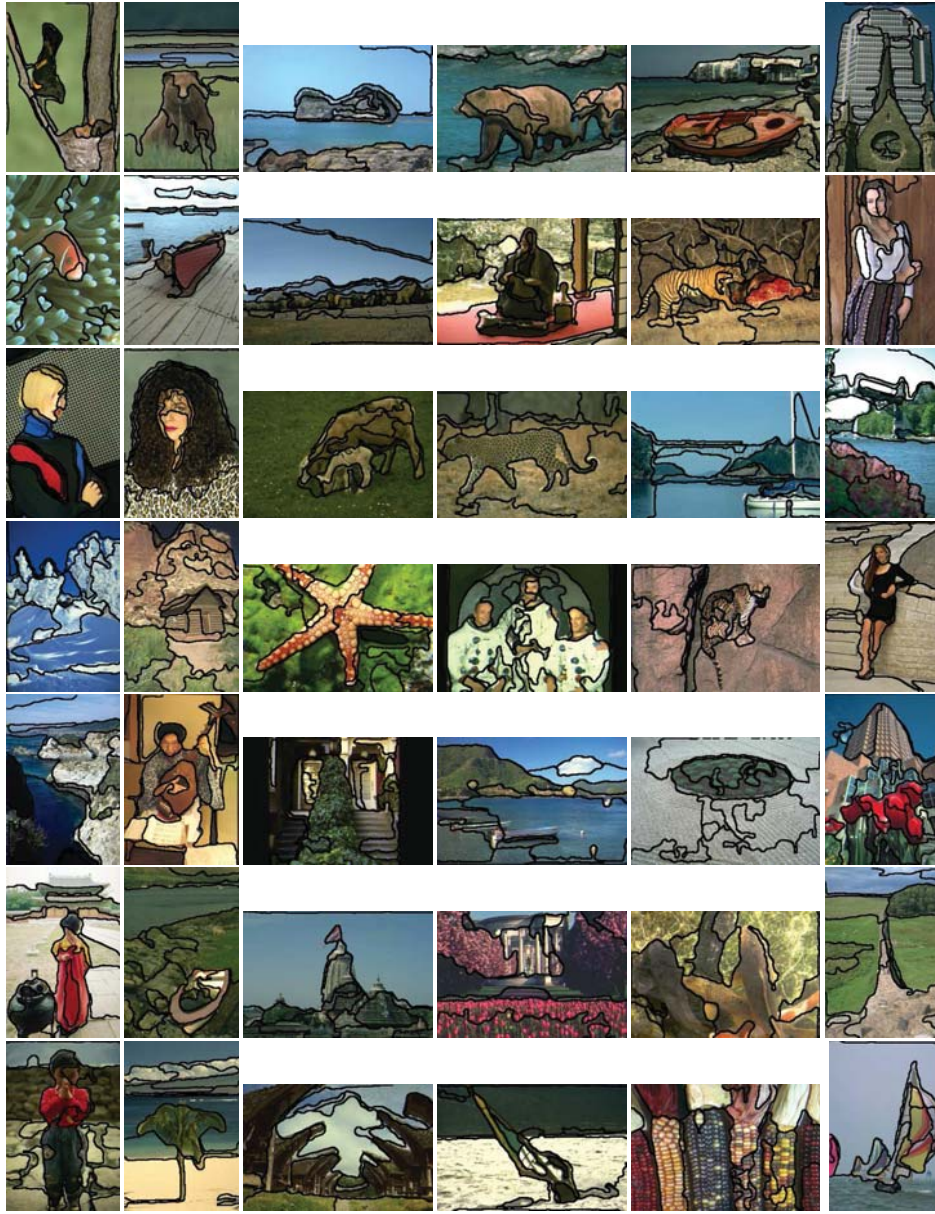


Fig. 6. Example of segmentations obtained by our algorithm  $\text{MD2S}_{[K=11|\zeta=0.40]}^{K\text{-means}}$  on several images of the Berkeley image database (see also Table I for quantitative performance measures and <http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/md2s.html> for the segmentation results on the entire database).

measure, which seems to be also highly correlated with human hand-segmentations [19] (a score equal to  $\text{PRI} = 0.78$ , for example, simply means that, on average, 78% of pairs of pixel labels are correctly classified in the segmentation results).

#### D. Results and Discussion

Table I shows the obtained PRI results for the different algorithms. Fig. 5 shows the distribution of the PRI measure over the 300 images of the Berkeley image database. To be impartial and for comparison, we illustrate the results of our simplest segmentation algorithm  $\text{MD2S}_{[K=11|\zeta=0.4]}^{K\text{-means}}$  by showing the same segmented images (see Figs. 6 and 7) as those shown in [29], in which the best existing segmentation

algorithm, in the PRI score sense, (see Table I) is described. The results for the entire database are available online at <http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/md2s.html>.

It may be noted that our segmentation procedures give competitive PRI results among the state-of-the-art segmentation methods recently proposed in the literature even with a final simple clustering scheme based on a  $K$ -means algorithm and a simple texture feature extraction step based on the values of the local requantized color and gradient distributions. This nonlinear dimension reduction pretreatment also allows to significantly improve the mean-shift segmentation results by increasing its PRI score from 0.755 to 0.786 (see Table I). This is mainly because there is a real interest to learn a nonlinear low-dimensional representation of the texture feature set for the segmentation problem. The same clustering scheme

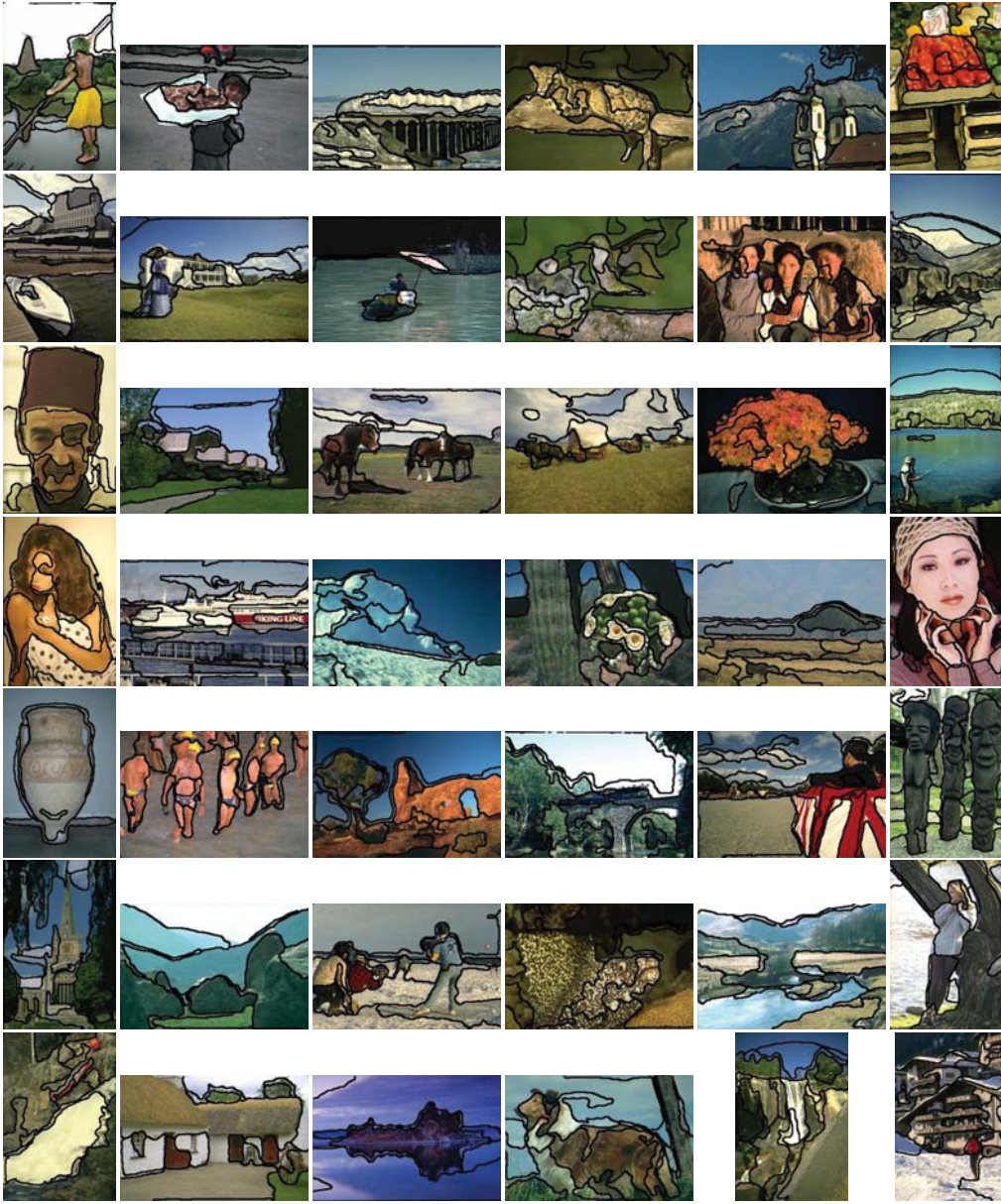


Fig. 7. Example of segmentations obtained by our algorithm  $\text{MD2S}_{[K=11|\xi=0.40]}^{K\text{-means}}$  on several images of the Berkeley image database (see also Table I for quantitative performance measures and <http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/md2s.html> for the segmentation results on the entire database).

( $K$ -means) without dimensionality reduction yields a significantly lower PRI score of around  $\text{PRI} = 0.72$  and  $\text{PRI} = 0.73$  with a classical PCA-based reduction method (in this case, the PCA does not provide more textural information but seems to just simplify a bit the clustering problem). Some textured images are very hard to segment because of the camouflaging property of the texture of some animals, but become easy to segment after our nonlinear dimensionality reduction step. This leads us to think that this type of camouflage texture data lies on an embedded nonlinear manifold within the higher dimensional space, which then makes the clustering or optimization algorithms subsequently used in segmentation less efficient. However, by comparison to [29], the segmentation results obtained are a bit oversegmented. Consequently, our segmentation procedure should provide better results if a final

grouping at region level (including or not an *a priori* on the number and/or the shape of these segmented regions) would be used as post-treatment.

Let us also mention that these oversegmented images induce some false contours (see Figs. 6 and 7). In order to quantify this problem and to measure the contour accuracy of our segmentation algorithm, we have computed the boundary displacement error (BDE) distance measure (the lower the distance, the better) proposed by [68] on the entire Berkeley database. The BDE measures the average displacement error of boundary pixels between two segmented images. Particularly, it defines the error of one boundary pixel as the distance between the pixel and the closest pixel in the other boundary image. For our algorithm, we obtain  $\text{BDE} = 10.3$  for  $\text{MD2S}_{[K=11|\xi=0.4]}^{K\text{-means}}$  and  $\text{BDE} = 9.8$  for



MD2S<sub>mean-shift</sub><sub>[ $\sigma_s=27|\sigma_r=55|\zeta=0.4$ ]</sub> compared to 9.9 for algorithm CTM [19] and 10.0 for algorithm FH (result available in [19]), which, in terms of contour accuracy is nearly comparable, but, as mentioned earlier, this boundary-based performance measure could be greatly improved in our case by adding a final merging step in the segmentation procedure.

### E. Comparison with FastMap

We have also compared our multiresolution MDS strategy with another recent scalable MDS technique, namely the FastMap algorithm [46] (see also Section I). It is worth pointing out that it was reported in [51] that the FastMap algorithm is as efficient as the best existing scalable MDS, the so-called Landmark MDS, in term of accuracy and speed comparisons, for very low embedding dimensions (i.e.,  $d \leq 2$ ), which is the case in our image segmentation application. The quality of the FastMap and our multiresolution MDS is measured in four different ways: 1) first by visual inspection of the low 3-D (detextured) representation (as a color image) of the input image [see Fig. 3]; 2) as in [51], by comparing the root mean square (RMS) relative distance error (RDE) introduced by the embedding; 3) by comparing the PRI-based segmentation score obtained by a  $K$ -means clustering on each of the two detextured images on the Berkeley database (after our learning step); and 4) by the CPU time. To this end, we have thus modified the FastMap algorithm (and the C source code, available online from the author's site [46]) so that it runs with our (non-Euclidean) Bhattacharyya distance [see (2)] along with the heuristic proposed in [47] for non-Euclidean distance metric (which prevents projections with nonpositive distance values at each iteration of the FastMap procedure). On the other hand, our spatial regularizing prior  $w_{s,t}$  (equal to 0.5 for sites  $t$  belonging to the first neighborhood of  $s$  [see (2)]) could not be included in the FastMap algorithm since the algorithm does not exploit the lattice structure of an image data. Fig. 1 shows the detextured image given by the FastMap algorithm. Contrary to our approach, we can easily see that the FastMap-based detexturing approach converts the different textural regions of the original image with less uniform colors (e.g., the girl's face or sweater) and seems to wrongly use, in the third image, similar colors for two different textured region (e.g., the red peppers and the foreground region of this image). On the other hand, the FastMap is 10 times faster of our multiresolution MDS (5 versus 50 s).

Let us now compare the accuracy in terms of the RMS-RDE introduced by the embedding. As in [51], for the FastMap whose the stress or cost functions is  $\sum_{s,t} (\hat{d}_{s,t} - d_{s,t})^2$ , this RMS-RDE is defined by

$$\text{RMS}_{\text{RDE}} = \sqrt{\frac{1}{N(N-1)} \sum_{s,t} \left( \left[ \frac{\rho \hat{d}_{s,t}}{d_{s,t}} \right] - 1 \right)^2} \quad (6)$$

with

$$\rho = \frac{\sum_{s,t} \hat{d}_{s,t} / d_{s,t}}{\sum_{s,t} \hat{d}_{s,t}^2 / d_{s,t}^2} \quad (7)$$

with  $d_{s,t}$  and  $\hat{d}_{s,t}$ , respectively, the true (unembedded) and the estimated (in the embedded space) distance, and  $\rho$  a scaling

factor.  $\sum_{s,t}$  integrates to the  $N(N-1)$  pairs of sites existing in the image. For our multiresolution MDS, whose the cost functions is  $\sum_{s,t} (\hat{d}_{s,t}^2 - w_{s,t} d_{s,t})^2$ , we have also used the RMS-RDE on this (slightly different) cost function

$$\text{RMS}_{\text{RDE}} = \sqrt{\frac{1}{N(N-1)} \sum_{s,t} \left( \left[ \frac{\rho \hat{d}_{s,t}^2}{(w_{s,t} d_{s,t})} \right] - 1 \right)^2} \quad (8)$$

with

$$\rho = \frac{\sum_{s,t} \hat{d}_{s,t}^2 / (w_{s,t} d_{s,t})}{\sum_{s,t} \hat{d}_{s,t}^4 / (w_{s,t} d_{s,t})^2}. \quad (9)$$

Since we have, for each detextured image, two embeddings [see (1)] and twice as much weight for the color-cue-based mapping compared to the gradient-feature-based mapping, we have weighted accordingly the two  $\text{RMS}_{\text{RDE}}$  measures. We have computed the average  $\text{RMS}_{\text{RDE}}$  on the whole Berkeley image database (with a subsampling factor of 10 pixels in length and width) and obtained  $\text{RMS}_{\text{RDE}} = 0.9441$  for the FastMap and  $\text{RMS}_{\text{RDE}} = 0.9244$  for our multiresolution MDS, which is slightly better in the case of the multiresolution MDS. Nevertheless, it remains difficult to quantify the difference in performance of these two scalable MDS techniques only on the values of the RMS-RDE metrics for two reasons. The first is due to the fact that we have two slightly different cost functions, the second is that the error can be very different in nature. More precisely, identical error in the RMS-RDE metric sense may affect differently the segmentation procedure. In our multiresolution MDS scheme, the presence of the regularizing factor  $w_{s,t}$  and the contextual interpolation scheme constrain the mapping, i.e., the detextured image, to be homogeneous which, in our case, greatly simplifies the subsequent segmentation procedure. Our regularizing factor thus incorporates knowledge concerning the types of detextured images (*a priori* defined as interesting solutions) but may introduce errors in the resulting low-dimensional mapping, errors which are reflected in the RMS-RDE metrics (but which can surely help the segmentation procedure). For that reason, we have to learn the internal parameters of the FastMap +  $K$ -means segmentation strategy by our grid search technique and finally compare the resulting PRI score. In this case, we obtain a lower PRI score around  $\text{PRI} = 0.755$  with the FastMap +  $K$ -means segmentation algorithm (see Table I). In our opinion, there are possibly several explanations for this (whose order of importance remains quite difficult to determine).

- 1) This may be because our non-Euclidean Bhattacharyya distance (used between each sample pair) creates for the FastMap (and also for all MDS methods based on eigendecomposition) a non-Euclidean distance matrix and thus a Gramian matrix which is no more positive semidefinite (PSD) and consequently with possible negative eigenvalues, thus disturbing the convergence and accuracy of the FastMap algorithm [51] (this is even more important because the distance used in our case is non-linear). As explained in [69], this is somewhat equivalent to considering a noise corrupted version of the Gramian matrix. It is also worth out that, with non-Euclidean distance matrix, eigendecomposition-based MDS is not

equivalent to directly optimizing over distance matrices [69]. These latter methods somewhat circumvent this by using the heuristic of replacing all negative eigenvalues with a null value, which is equivalent to approximating the original distance matrix (in the sense of Frobenius norm) and finally finding an embedding solution [69], [70]. Another strategy that makes the Gramian matrix to be PSD is the recent constant-adding method adopted in [71], which consists in finding an appropriate constant to be added to all dissimilarity distances (apart from the self-dissimilarities) such that the distance kernel matrix is guaranteed to be PSD. This constant can be estimated by the largest eigenvalue of a matrix built from the input distance matrix. Since this constant-adding strategy requires preliminary computation of the distance matrix associated with all pairs of points, it is worth noting that this method cannot be applied, in any obvious way, to (scalable) Landmark MDS or FastMap algorithms.

- 2) This is also due to the very scalable method used in the FastMap [46]. Indeed, the heart of the FastMap is based on the Pythagorean theorem and a sequence of several projections onto hyperplanes perpendicular to pivot axes (defined by distant pairs of samples). As explained in [65], the Pythagorean theorem is specific to Euclidean distances and does not hold if non-Euclidean distances are used. The use of a non-Euclidean distance in such a method (such as the Bhattacharyya distance) may cause the pruning property to be violated and this can severely degrade the performance of FastMap. In order to somewhat alleviate this problem, a heuristic is proposed in [47] (and this heuristic was included in our FastMap algorithm code). Nevertheless, as explained in [65], it resolves the drawback of negative projected distances but does not correct the fundamental problem, namely the fact that non-Euclidean distances do not satisfy the triangle inequality and the Pythagorean theorem.
- 3) It may also be due to the spatial regularizing factor ( $w_{s,t}$ ) used in our multiresolution MDS method, and which helps to estimate a less noisy color (de-textured) image by simply modeling the fact that spatial neighboring samples are likely to belong to the same region (and which then facilitates the subsequent  $K$ -means-based segmentation procedure).
- 4) It is also possible that our interpolation scheme (see Section IV-B), which also exploits the inherent spatial dependencies between neighbouring texture feature vectors and the weighted average formula (also efficiently used in competitive denoising procedures such as the non-local means filtering strategy [62]), acts as a regularizing factor which efficiently constrains (thus, especially for image data) our MDS dimensionality reduction model and the subsequent  $K$ -means clustering procedure.

For comparison, in terms of computational complexity, the cost of running Landmark (ISOMAP) MDS algorithm with  $N$  points,  $n$  landmarks, and  $p$  nearest neighbors in the (target) low-dimensional  $d$  space is  $O(Nnp + Nnd + n^3)$  if the

neighborhood graph is given [48]. The cost of building the  $p$ -nearest neighbor graph has complexity  $O(N^2)$ . Nevertheless, as indicated in [48], there are more sophisticated techniques that give an expected bound of  $c^{O(1)}(Np + N \log \log n)$  for datasets with an expansion coefficient of  $c$ . For data sampled randomly from a  $d$ -dimensional sub-manifold (of only Euclidean space), the expansion coefficient is  $c = 2^d$ .

The computational complexity of FastMap is  $O(ND^2)$ , where we recall that  $D$  is the dimensionality of the high-dimensional representation. Indeed, in this model, the computation is dominated by  $D$  deflations, i.e., concretely  $D$  projections onto hyperplanes, each operating on  $N$  data samples and each of which takes  $O(D)$  operations [51]. It is also shown in [51] that the CPU times between these two scalable MDS algorithms remain comparable.

In comparison, the time complexity of our algorithm, with two levels of resolution (i.e., the full resolution and an image 16 times smaller in length and width [see Section IV]), is  $O(NN_s^2 D + (N/16^2)N_s^2[L_{\max}^D + L_{\max}^S] + NN_s^2 L_{\max}^S + NN_I^2 D)$  for an image with  $N$  pixels. More precisely,  $O(NN_s^2 D)$  is the time complexity of the computation of the set of distances at the full and coarser resolution levels.  $N_s$  is the size of the square neighborhood window ( $N_s = 15$  in our algorithm) and  $D$  is the dimensionality of the high-dimensional representation.  $O((N/16^2)N_s^2[L_{\max}^D + \max])$  and  $O(NN_s^2 L_{\max}^S)$  are respectively the time complexity of the hybrid and stochastic optimizations at the full and coarser resolution levels, with  $L_{\max}^D$  and  $L_{\max}^S$  representing the maximum number of iterations of the conjugate gradient and Metropolis searches ( $L_{\max}^D$  and  $L_{\max}^S < 100$  in our algorithm since these iterative optimizers can stop before the maximum number of iterations, when convergence is reached). Finally,  $O(NN_I^2 D)$  is the time complexity of our interpolation scheme, with  $N_I$  denoting the size of the squared neighborhood window used in our interpolation scheme ( $N_I = 5$  in our algorithm). Since  $D$ ,  $N_s$ ,  $N_I$ ,  $L_{\max}^D$ , and  $L_{\max}^S$  are much lower than  $N$ , our algorithm can be implemented in a linear polynomial time as the Landmark (except for the building of the nearest neighborhood graph of the Landmark algorithm which should use a non-Euclidean distance to be compared with our Bhattacharyya-based reduction model) and the FastMap algorithms. In terms of computing time, as previously said, the FastMap is 10 times faster than our multiresolution MDS, more precisely 5 versus 50 s for a  $320 \times 240$  image (and the computing time of the Landmark is comparable to the FastMap algorithm when Euclidean distances are used [51]).

#### F. Algorithm

The segmentation procedure takes about 1 min (on average, 50 s for the dimensionality reduction and approximately 5 – 10 s for the clustering per image for an i7-930 Intel CPU, 2.8 GHz, 5611 bogomips and for a nonoptimized code running on Linux for the entire segmentation procedure. Source code (in C++ language) of our algorithm (with the set of segmented images) are publicly available at the following <http://www.iro.umontreal.ca/~mignotte/ResearchMaterial/md2s.html>, in order to make possible comparisons with



future segmentation algorithms or different performance measures.

## VI. CONCLUSION

In this paper, we have proposed a robust multiresolution framework for reliable solution of MDS-based nonlinear and large-scale dimensionality reduction problems. As application, we have demonstrated the interest of this algorithm to learn a nonlinear low-dimensional representation of the set of high-dimensional textural features of an image. In our application, this low-dimensional representation is a nonlinear embedding in which the local color value distribution of the texture region is approximated by pairwise pixels whose values is separated by Bhattacharyya distances. This dimension reduction step, which can be viewed as a detexturing approach that converts the original texture image into a noisy image with regions with uniform colors in a low-dimensional space, simplifies greatly the segmentation problem of textured images and appears to be an interesting alternative to complex segmentation models existing in the literature.

## ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their many valuable comments and suggestions that helped to improve both the technical content and the presentation quality of this paper.

## REFERENCES

- [1] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.
- [2] F. Destremes, M. Mignotte, and J.-F. Angers, "A stochastic method for Bayesian estimation of hidden Markov random field models with application to a color model," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 1096–1108, Aug. 2005.
- [3] R. Hedjam and M. Mignotte, "A hierarchical graph-based Markovian clustering approach for the unsupervised segmentation of textured color images," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 1365–1368.
- [4] S. P. Chatzis and G. Tsechenakis, "The infinite hidden Markov random field model," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 1004–1014, Jun. 2010.
- [5] E. Csemeli and D. Wang, "Texture segmentation using Gaussian-Markov random fields and neural oscillator networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 394–404, Mar. 2001.
- [6] C. Nikou, A. C. Likas, and N. P. Galatsanos, "A Bayesian framework for images segmentation with spatially varying mixtures," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2278–2289, Sep. 2010.
- [7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [9] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan, "Normalized tree partitioning for image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, Jun. 2008, pp. 1–8.
- [10] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 925–936, Jul. 2005.
- [11] K. Chen, D. Wang, and X. Liu, "Weight adaptation and oscillatory correlation for image segmentation," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1106–1123, Sep. 2000.
- [12] P. A. Arbeláez and L. D. Cohen, "A metric approach to vector-valued image segmentation," *Int. J. Comput. Vis.*, vol. 69, no. 1, pp. 119–126, Aug. 2006.
- [13] M. Krimidis and I. Pitas, "Color texture segmentation based on the modal energy of deformable surfaces," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1613–1622, Jul. 2009.
- [14] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [15] Q. Luo and T. M. Khoshgoftaar, "Unsupervised multiscale color image segmentation based on MDL principle," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2755–2761, Sep. 2006.
- [16] M. A. Carreira-Perpiñán, "Fast nonparametric clustering with Gaussian blurring mean-shift," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, Jun. 2006, pp. 153–160.
- [17] M. Mignotte, "Segmentation by fusion of histogram-based  $K$ -means clusters in different color spaces," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 780–787, May 2008.
- [18] D. E. Ilea and P. F. Whelan, "CTex—An adaptive unsupervised segmentation algorithm on color-texture coherence," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1926–1939, Oct. 2008.
- [19] A. Y. Yang, J. Wright, S. S. Sastry, and Y. Ma, "Unsupervised segmentation of natural images via lossy data compression," *Comput. Vis. Image Understand.*, vol. 110, no. 2, pp. 212–225, May 2008.
- [20] A. Diplaros, N. Vlassis, and T. Gevers, "A spatially constrained generative model and an EM algorithm for image segmentation," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 798–808, May 2007.
- [21] M. M. Mushrif and A. K. Ray, "A-IFS histon based multithresholding algorithm for color image segmentation," *IEEE Signal Process. Lett.*, vol. 16, no. 3, pp. 168–171, Mar. 2009.
- [22] G. Sfikas, C. Nikou, and N. Galatsanos, "Edge preserving spatially varying mixtures for image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, vol. 1. Anchorage, AK, Jun. 2008, pp. 1–7.
- [23] A. P. Benavent, F. E. Ruiz, and J. M. Saez, "Learning Gaussian mixture models with entropy-based criteria," *IEEE Trans. Neural Netw.*, vol. 20, no. 11, pp. 1756–1771, Nov. 2009.
- [24] K. K. Bhoyar and O. G. Kakde, "Color image segmentation based on JND color histogram," *Int. J. Image Process.*, vol. 3, no. 6, pp. 282–293, Jan. 2010.
- [25] I. Mecimore and C. D. Creusere, "Unsupervised bitstream based segmentation of images," in *Proc. 5th IEEE Signal Process. Educ. Workshop Dig. Signal Process., Workshop*, Marco Island, FL, Jan. 2009, pp. 643–647.
- [26] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
- [27] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.
- [28] F. Destremes, J.-F. Angers, and M. Mignotte, "Fusion of hidden Markov random field models and its Bayesian estimation," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2920–2935, Oct. 2006.
- [29] M. Mignotte, "A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1610–1624, Jun. 2010.
- [30] S. Ghosh, J. J. Pfeiffer, and J. Mulligan, "A general framework for reconciling multiple weak segmentations of an image," in *Proc. Conf. IEEE Workshop Appl. Comput. Vis.*, Snowbird, UT, Dec. 2009, pp. 1–8.
- [31] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Proc. Asian Conf. Comput. Vis.*, Part I, LNCS 5994, Sep. 2009, pp. 135–146.
- [32] H. Du, H. Qi, X. Wang, and R. Ramanath, "Band selection using independent component analysis for hyperspectral image processing," in *Proc. 32nd Appl. Imagery Pattern Recog. Workshop*, Washington D.C., Oct. 2003, pp. 93–98.
- [33] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, Jun. 2007, pp. 1–6.
- [34] A. Mohan, G. Sapiro, and E. Bosh, "Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 2, pp. 206–210, Apr. 2007.
- [35] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 441–454, Mar. 2005.
- [36] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

- [37] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2001, pp. 585–591.
- [38] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci.*, vol. 100, no. 10, pp. 5591–5596, Apr. 2003.
- [39] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 148–154, Jan. 1997.
- [40] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
- [41] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [42] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. London, U.K.: Chapman & Hall, 1994.
- [43] G. Young and A. Householder, "Discussion of a set of points in terms of their mutual distances," *Psychometrika*, vol. 3, no. 1, pp. 19–22, Mar. 1938.
- [44] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.
- [45] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. R. Barra, F. Brodeau, G. Romier, B. Van Cutsem, Eds. Amsterdam, The Netherlands: North Holland, 1977, pp. 133–145.
- [46] C. Faloutsos and K.-I. Lin, "Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, San Jose, CA, May 1995, pp. 163–174.
- [47] J. T.-L. Wang, X. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang, "Evaluating a class of distance-mapping algorithms for data mining and clustering," in *Proc. 5th Int. Conf. ACM SIGKDD*, 1999, pp. 307–311.
- [48] V. de Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Dept. Math., Stanford University, Stanford, CA, Tech. Rep., Jun. 2004.
- [49] S. Ketprechasawat and O. C. Jenkins, "Hierarchical landmark charting," M.S. thesis, Dept. Comput. Sci., Brown University, Providence, RI, 2006.
- [50] S. Lee and S. Choi, "Landmark MDS ensemble," *Pattern Recog.*, vol. 42, no. 9, pp. 2045–2053, Sep. 2009.
- [51] J. C. Platt, "Fastmap, metricmap, and landmark MDS are all Nyström algorithms," in *Proc. 10th Int. Workshop Artif. Intell. Stat.*, Jan. 2005, pp. 261–268.
- [52] T. Yang, J. Liu, L. McMillan, and W. Wang, "A fast approximation to multidimensional scaling," in *Proc. ECCV-Workshop Comput. Intensive Methods Comput. Vis.*, 2006, pp. 1–8.
- [53] Y. Chen, L. Wang, M. Dong, and J. Hua, "Exemplar-based visualization of large document corpus," *IEEE Trans. Visual. Comput. Graph.*, vol. 15, no. 6, pp. 1161–1168, Nov.–Dec. 2009.
- [54] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Stat. Soc.*, vol. 48, no. 3, pp. 259–302, 1986.
- [55] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
- [56] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.
- [57] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on trees: Message-passing and linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.
- [58] R. Szeliski, R. Zabih, D. Sharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.
- [59] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, Sep. 1990.
- [60] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy, "Sonar image segmentation using an unsupervised hierarchical MRF model," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1216–1231, Jul. 2000.
- [61] M. Mignotte, "Nonparametric multiscale energy-based model and its application in some imagery problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 184–197, Feb. 2004.
- [62] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
- [63] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [64] F. Murtagh and A. Heck, *Multivariate Data Analysis with Fortran, C and Java Code*. Dordrecht, The Netherlands: Kluwer, 1987.
- [65] G. R. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 530–549, May 2003.
- [66] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int. Conf. Comput. Vis.*, vol. 2. Vancouver, BC, Canada, Jul. 2001, pp. 416–423.
- [67] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "A measure for objective evaluation of image segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, vol. 3. San Diego, CA, Jun. 2005, pp. 34–41.
- [68] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. 7th Eur. Conf. Comput. Vis.*, Copenhagen, Denmark, May 2002, pp. 408–422.
- [69] L. Cayton and S. Dasgupta, "Robust Euclidean embedding," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, Jun. 2006, pp. 169–176.
- [70] M. W. Trosset and C. E. Priebe, "The out-of-sample problem for classical multidimensional scaling," *Comput. Stat. & Data Anal.*, vol. 52, no. 10, pp. 4635–4642, Jun. 2008.
- [71] H. Choi and S. Choi, "Robust kernel isomap," *Pattern Recog.*, vol. 40, no. 3, pp. 853–862, Mar. 2007.



**Max Mignotte** received the D.E.A. (postgraduate) degree in digital signal, image, and speech processing from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1993, and the Ph.D. degree in electronics and computer engineering from the University of Bretagne Occidentale, Brest, France, and the Digital Signal Laboratory, French Naval Academy, Lanvéoc, France, in 1998.

He was with the National Institute for Research in Computer Science and Control Post-Doctoral Fellow at the Département d'Informatique et de Recherche Opérationnelle (DIRO), University of Montreal, Montreal, QC, Canada, from 1998 to 1999. He is currently with DIRO at the Computer Vision & Geometric Modeling Laboratory, University of Montreal, as an Associate Professor. His current research interests include statistical methods, Bayesian inference, and hierarchical models for high-dimensional inverse problems.

Prof. Mignotte is a member of the Laboratoire de Recherche en Imagerie et Orthopédie, Research Center of the Centre Hospitalier de l'Université de Montréal (CHUM), Hôpital Notre-Dame, and a Researcher at CHUM.